BIGDATA Processing with Hadoop
(MR Programming)

Problem 1: Grep the word

Write a MapReduce program to output only the lines that contain the word "torture" in a given text file. You can use any text file as test input for your program.

Problem 2: Most frequent Bigram

Write a MapReduce program to output only the single most common bigram (pair of adjacent words) in the dataset. Split words on whitespace. You can find the test dataset at following location:
https://github.com/algorithmica-repository/hadoop-bigdata/datasets/bible.txt

Problem 3: Finding the mean temperature by month

Write a MapReduce program that computes the mean max temperature for every month. You can find the test dataset at following location:
https://github.com/algorithmica-repository/hadoop-bigdata/datasets/temperature.csv
The fields of temperature data set are: DDMMYYYY, MIN_TEMP, MAX_TEMP
HINT: mean(a,b,c,d,e) != mean(mean(a,b,c), mean(d,e)) where as
sum(a,b,c,d,e) = sum(sum(a,b,c),sum(d,e))

Problem 4: Finding Top-N used words

Write a MapReduce program that finds the top-20 most used words in a given text file. You can find the test dataset at following location:
https://github.com/algorithmica-repository/hadoop-bigdata/datasets/bible.txt

Problem 5: Most frequent Vowel and Consonant words

Write a MapReduce program that outputs the most common word that starts with a vowel and the most common word that starts with a consonant. The output should also include the number of times the words appear. You can find the test dataset at following location:
https://github.com/algorithmica-repository/hadoop-bigdata/datasets/bible.txt

Problem 6: Efficient Reduce-side Join

In class we discussed one implementation of Reduce-side join in which we generated unique id for each dataset as part of value in map-phase. We can make it more efficient by generating the unique id as part of key itself. Think of an MR algorithm with this change and modify the code given in the class and compare the performance of both strategies.